



HAL
open science

Online greedy identification of linear dynamical systems

Matthieu Blanke, Marc Lelarge

► **To cite this version:**

Matthieu Blanke, Marc Lelarge. Online greedy identification of linear dynamical systems. CDC 2022 - 61st Conference on Decision and Control, IEEE, Dec 2022, Cancun, Mexico. pp.5363-5368, 10.1109/CDC51059.2022.9993030 . hal-03938609

HAL Id: hal-03938609

<https://hal.science/hal-03938609>

Submitted on 13 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Online greedy identification of linear dynamical systems

Matthieu Blanke and Marc Lelarge

Abstract—This work addresses the problem of exploration in an unknown environment. For multi-input multi-output, linear time-invariant dynamical systems, we use an experimental design framework and introduce an online greedy policy where the control maximizes the information of the next step. We evaluate our approach experimentally and compare it with more elaborate gradient-based methods. In a setting with a limited number of observations, our algorithm has low complexity and shows competitive performances.¹

I. INTRODUCTION

System identification is a problem of great relevance in many fields such as econometrics, robotics, aeronautics, mechanical engineering or reinforcement learning [1]–[5]. The task consists in estimating the parameters of an unknown system by sampling data from it. One is often interested in identifying the system with as few samples as possible, hence motivating the active identification paradigm: the controller wants to choose the inputs that yield maximally informative data. We focus on multi-input multi-output (MIMO) linear time-invariant (LTI) dynamical systems, which can model for example the motion of an aircraft system over time near an equilibrium position. Let $A \in \mathbb{R}^{d \times d}$ and $B \in \mathbb{R}^{d \times m}$ be two matrices; we consider the following discrete-time dynamics:

$$\begin{aligned} x_0 &= 0, \\ x_{t+1} &= Ax_t + Bu_t + w_t, \quad 0 \leq t \leq T-1, \end{aligned} \quad (1)$$

where $x_t \in \mathbb{R}^d$ is the state vector, $w_t \sim \mathcal{N}(0, \sigma^2 I_d)$ is a normally distributed isotropic noise with known variance σ^2 and the control variables $u_t \in \mathbb{R}^m$ are chosen by the controller with the following power constraint:

$$\frac{1}{T} \sum_{t=0}^{T-1} \|u_t\|^2 \leq \gamma^2. \quad (2)$$

The parameters of the system are $(A B) := \theta \in \mathbb{R}^{d \times q}$ ($q = d + m$). We note $(A_* B_*) := \theta_*$ the parameters of the real system which are unknown initially and are to be identified. It may happen that the controller knows B_* in advance, in which case $\theta = A$ and $q = m$. At time T , the controller returns an estimate of θ_* from the observed trajectory $(x_t)_{0 \leq t \leq T}$. The goal of active system identification is to choose inputs (u_t) that make the trajectory as informative as

possible for the estimation of θ_* , with input u_t being chosen at time t using the past observations. We provide a formal mathematical formulation of this objective in Section I-D.

In practice, one must face two main limitations in the identification of a LTI dynamical system. First, the number of observations T is small. Indeed systems have complex dynamics in general and can only be approximated by LTI systems on short time scales. Moreover, running an experiment on the real system can be costly (think of an aircraft test flight for instance), so the controller wants to estimate the parameters with as few samples as possible. Second, our identification algorithm needs to take decisions as fast as possible in order to run realistically online. Hence, we attach great importance to the computational time C of the control design, which should be reasonably small for practical applications. We have in this work a critical eye on these two limitations and we seek an exploration algorithm that runs fast and is sample efficient in a regime with a limited number of observations.

A. Contributions

To the best of our knowledge, system identification guarantees are only available in the large T limit, making the hypothesis of linear time-invariant dynamics quite unlikely. Instead, we explore in this work a practical setting for linear system identification with strong limitations on the number of interactions with the real system and on the computational resources used for input design and estimation. Using a framework based on experimental design, we propose an online greedy algorithm requiring minimal resources. The resulting policy gives a control that maximizes the amount of information collected the next step. We show empirically that for short interactions with the system, this simple approach can actually outperform more sophisticated gradient-based methods. We study the computational complexity of our algorithm and compare its performance with gradient-based approaches and with an oracle that we have designed, both on average and on real-life dynamical systems.

B. Related work

System identification is a primary area of control theory [1], [6]. It has been widely studied in the field of experimental design. The question of choosing the most informative input can be tackled in the framework of classical experimental design [7], [8]. For the particular case of LTI dynamical systems, optimal design approaches have provided results for single-input single-output (SISO) systems [3], [9] or MIMO systems in the frequency domain or with randomized inputs [10].

This work was partially supported by the French government under management of Agence Nationale de la Recherche as part of the “Investissements d’avenir” program, reference ANR19-P3IA-0001 (PRAIRIE 3IA Institute).

M. Blanke and M. Lelarge are with INRIA, DI/ENS, PSL Research University, Paris, France. matthieu.blanke@inria.fr, marc.lelarge@inria.fr

¹Our code is available at <https://github.com/MB-29/greedy-identification>

More recently, system identification has received considerable attention in the machine learning community, with the aim of obtaining finite-time bounds on the estimation error [11]–[14]. The issue of designing optimal inputs is tackled in [15], [16]. The authors derive an asymptotically optimal algorithm (as the number of observations tends to infinity) by maximizing an optimal design objective in the frequency domain. The inputs are played and updated in epochs with exponentially growing length.

C. Notations and assumptions

In the rest of this work, we note $\theta_\star = (A_\star B_\star)$ the unknown parameters underlying the dynamics. We define a policy $\pi : (x_{1:t}, u_{0:t-1}) \rightarrow u_t$ as a mapping between the past trajectory and the future input. The set of policies satisfying the power constraint (2) is noted Π_γ . We denote $\tau = (x_{1:T}, u_{0:T-1})$ a trajectory, and we extend this notation to $\tau(\pi, T)$ when the trajectory is obtained using a policy π until time T . We denote \mathbb{E}_θ the average for a dynamical system given by (1) with fixed $(A B) = \theta$ knowing the past trajectory, where the expectation is taken over the randomness of the noise w_t and possibly from the policy inducing the control u_t . We assume that the pair (A_\star, B_\star) is controllable: the matrix $R_\star = (B_\star A_\star B_\star \dots A_\star^{d-1} B_\star)$ has rank d .

D. Adaptive identification

Let us fix an estimator $\hat{\theta} : \tau \mapsto \hat{\theta}(\tau) \in \mathbb{R}^{d \times q}$, yielding an estimate of the parameters from a given trajectory. Our objective is to find a policy $\pi \in \Pi_\gamma$ whose resulting trajectory $\tau(\pi, T)$ gives a good estimate $\hat{\theta}(\tau)$ for θ_\star . We measure this performance by the mean squared error of the parameters:

$$\text{MSE}(\pi) = \frac{1}{2} \mathbb{E}_{\theta_\star} \left[\left\| \hat{\theta}(\tau(\pi, T)) - \theta_\star \right\|_F^2 \right]. \quad (3)$$

An optimal policy π_\star would be one minimizing the MSE (3). Of course, this quantity depends on θ_\star the true unknown parameter of the system through the dynamics (1), so finding π_\star by computing and minimizing (3) is not possible. However, we can adaptively compute an approximation of the MSE by estimating θ_\star sequentially, as follows.

Definition 1 (Adaptive system identification): Given an estimate $\hat{\theta}_i$ of θ_\star , the next sequence of inputs can be chosen to minimize a cost function F approximating the MSE (3), using $\hat{\theta}_i$ as an approximation of θ_\star . Then, these inputs are played and θ_\star is re-estimated with the resulting trajectory yielding an estimate $\hat{\theta}_{i+1}$, and so on. We call planning the computational process of choosing the next inputs by optimizing the cost F . As observations are collected, the estimation of θ_\star gets more and more accurate, hence the planning functional becomes closer and closer to the MSE, yielding an increasingly accurate policy with each iteration.

Adaptive identification is summarized in Algorithm 1, which takes as inputs a first guess for the parameters to estimate θ_0 , the parameters σ and γ of the problem, a planning schedule $\{t_0 = 0, t_1, \dots, t_{n-1}, t_n = T\}$ (n is the

number of planning iterations, *i.e.* the number of times the policy is updated between times 0 and T), a cost functional F and an estimator $\hat{\theta}$. An adaptive identification algorithm is

Algorithm 1 Sequential system identification

inputs initial guess θ_0 , noise variance σ^2 , power γ^2 , cost functional F , estimator $\hat{\theta}$, planning schedule $\{t_i\}_{0 \leq i \leq n}$
initialize π_0 returns random inputs
output final estimate θ_T
for $0 \leq i \leq n - 1$ **do**
 run the true system $t_{i+1} - t_i$ steps
 with inputs $u_t = \pi_i(x_{1:t}, u_{1:t-1})$
 $\theta_i = \hat{\theta}(x_{1:t_i}, u_{1:t_i-1})$ ▷ estimation
 π_i solves $\min_{\pi \in \Pi_\gamma} F(\pi; \theta_i, t_{i+1})$ ▷ planning
end for

hence determined by a triplet $(\hat{\theta}, F, \{t_i\})$. A natural estimator is the least squares estimator $\hat{\theta} = \theta_{\text{LS}}$ which we define in Section II-A. In the rest of this work, we set $\hat{\theta} = \hat{\theta}_{\text{LS}}$.

Example 1 (Random policy): A naive strategy for system identification consists in playing random inputs with maximal power at each time step. This corresponds to the choice $t_i = i$ and π_i returning $u_t \sim \mathcal{N}(0, \frac{\gamma^2}{m} I_m)$.

Example 2 (Task-optimal pure exploration): In [16], the authors propose for F an A-optimal design functional (see Section II-B for details) and show that it approximates the MSE (3) in the long time limit at an optimal rate when $T \rightarrow +\infty$. Their identification algorithm works with planning intervals of exponentially growing length $t_i = 2^i \times T_0$ for some initial duration T_0 .

Example 3 (Oracle): An oracle is a controller who is assumed to choose its policy with the knowledge of the true parameter θ_\star . Hence, it can perform one single, offline optimization of $F(\pi; \theta_\star, T) = \text{MSE}(\pi)$ over $\{t_i\} = \{0, T\}$. By definition, the inputs chosen by the oracle are the optimal inputs for our mean squared error system identification problem.

II. BACKGROUND

It is convenient to describe the structure of the state as a function of the inputs and noise. By integrating the dynamics (1), we obtain the following result.

Proposition 1: The state can be expressed as $x_t = \bar{x}_t + \tilde{x}_t$ with

$$\bar{x}_t = \sum_{s=0}^{t-1} A^{t-1-s} B u_s, \quad \tilde{x}_t = \sum_{s=0}^{t-1} A^{t-1-s} w_s. \quad (4)$$

Note that that $\bar{x}_t = \mathbb{E}_\theta[x_t]$ solves the deterministic dynamics $\bar{x}_{t+1} = A\bar{x}_t + B u_t$ and \tilde{x}_t has zero mean and is independent of the control. The two terms \bar{x}_t and \tilde{x}_t depend linearly on the $B u_s$ and the w_s respectively.

The data-generating distribution knowing the parameter θ can be computed using the probability chain rule with dynamics (1):

$$p(\tau|\theta) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^T \exp \left[-\frac{1}{2\sigma^2} \sum_{t=0}^{T-1} \|x_{t+1} - Ax_t - Bu_t\|_2^2 \right]. \quad (5)$$

We define the log-likelihood (up to a constant):

$$\begin{aligned} \ell(\tau, \theta) &= -\frac{1}{2\sigma^2} \sum_{t=0}^{T-1} \|x_{t+1} - Ax_t - Bu_t\|_2^2 \\ &= -\frac{1}{2\sigma^2} \|Y - Z\theta^\top\|_F^2, \end{aligned} \quad (6)$$

where we have noted $Y = (y_0 \dots y_{T-1})^\top \in \mathbb{R}^{T \times d}$ and $Z = (z_0 \dots z_{T-1})^\top \in \mathbb{R}^{T \times q}$ the observations and the covariates associated with the parameter θ . If $\theta = (A B)$, then $y_t = x_{t+1}$, $z_t = \begin{pmatrix} x_t \\ u_t \end{pmatrix}$. If $\theta = A$, then $y_t = x_{t+1} - Bu_t$ and $z_t = x_t$. We also note $U = (u_0 \dots u_{T-1})^\top \in \mathbb{R}^{T \times m}$, $X = (x_0 \dots x_{T-1})^\top \in \mathbb{R}^{T \times d}$ and $W = (w_0 \dots w_{T-1})^\top \in \mathbb{R}^{T \times d}$ the matrix representations of the input, the state and the noise. We define the moment matrix $M_t = \sum_{s=0}^{t-1} z_s z_s^\top$ and the Gramians of the system at time t : $\Gamma_t(\tau; \theta) = \frac{1}{t} \mathbb{E}_\theta [M_t]$. Note that $Z^\top Z = M_T$.

A. Ordinary least squares

Given the linear structure of the dynamics, a natural estimator for the matrix θ_* is the least squares estimator. The method of least squares provides us with a formula for the mean squared error with respect to the ground truth, which can be used as a measure of the quality of a control.

Proposition 2 (Ordinary least squares estimator): Given inputs U and noise W , the ordinary least squares (OLS) estimator associated with the resulting trajectory τ is

$$\hat{\theta}(\tau) = ((Z^\top Z)^{-1} Z^\top Y)^\top. \quad (7)$$

and its difference from θ_* is given by

$$\begin{aligned} (\hat{\theta}(\tau) - \theta_*)^\top &= (Z^\top Z)^{-1} Z^\top W \\ &= Z^+ W, \end{aligned} \quad (8)$$

where Z^+ denotes the pseudo-inverse of Z . Noting θ_t the least squares estimator obtained from the trajectory $\tau_t = (x_{0:t}, u_{0:t-1})$, we recall the recursive update formula

$$\theta_{t+1}^\top = M_{t+1}^{-1} (M_t \theta_t + z_t y_t^\top). \quad (9)$$

Definition 2 (OLS mean squared error): For a given trajectory τ generated with parameters θ_* and noise W , the Euclidean mean squared error (MSE) is

$$\begin{aligned} \|\hat{\theta}_{\text{LS}} - \theta_*\|_F^2 &= \|((Z^\top Z)^{-1} Z^\top W)^\top\|_2^2 \\ &= \text{tr} [Z (Z^\top Z)^{-2} Z^\top W W^\top]. \end{aligned} \quad (10)$$

If the noise W and the covariates Z were independent, the expected error would reduce to the A-optimal design objective $\mathbb{E}[\text{tr}(Z^\top Z)^{-1}]$. It is not the case in our framework since Z is generated with W .

B. Classical optimal design

The correlation between Z and W makes the derivation of a simple expression for the expectation of (10) challenging. In this section, we show how a more tractable objective can be computed by applying the theory of optimal design of experiments [7], [17]. In the classical theory of optimal design, the informativeness of an experiment is measured by the size of the expected Fisher information.

Definition 3 (Fisher information matrix): Let $\ell(\tau, \theta) = \log p(\tau|\theta)$ be the log-likelihood of the data distribution knowing the parameter θ . The Fisher information matrix is defined as

$$I(\theta) = -\mathbb{E}_\theta \left[\frac{\partial^2 \ell(\tau, \theta)}{\partial \theta^2} \right] \in \mathbb{R}^{qd \times qd}. \quad (11)$$

Proposition 3: For the LTI system (1),

$$I(\theta) = \frac{T}{\sigma^2} \text{diag}(\Gamma_T, \dots, \Gamma_T), \quad (12)$$

the number of blocks being d .

Proof: The log-likelihood (6) can be separated into a sum over the θ_j (d symmetric terms in θ_j). The quadratic term in θ_j is $\|Z\theta_j\|_2^2 = \theta_j^\top Z^\top Z \theta_j$ and the other terms are constant or linear. Differentiating twice and taking the expectation gives $\mathbb{E}_\theta[Z^\top Z]$, which yields the desired result after division by $-\sigma^2$. ■

Definition 4: In classical optimal design, the size of the information matrix is measured by some criterion $\Phi : \mathbb{S}_n^+(\mathbb{R}) \rightarrow \mathbb{R}_+$, which is a functional of its eigenvalues $\lambda_1, \dots, \lambda_d \geq 0$. The quantity $\Phi(I)$ represents the amount of information brought by the experiment and should be maximized.

Example 4: Some of the usual optimal design criteria include A-optimality $\Phi_A(I) = -\text{tr}(I^{-1})$ and D-optimality $\Phi_D(I) = \log \det I$.

The criteria have mathematical properties such as homogeneity, monotonicity and concavity in the sense of Loewner ordering, which can be interpreted in terms of information theory: monotonicity means that a larger information matrix brings a greater amount of information, concavity means that information cannot be increased by interpolation between experiments. We refer to [8] for more details.

The classical optimal design theory leads to the following definition of the optimal design planning functional.

Definition 5 (Optimal design functional): Let Φ denote an optimal design criterion. Then the associated cost is defined as

$$F_\Phi(\pi; \theta, t) = -\Phi [\Gamma_t(\tau(\pi); \theta)]. \quad (13)$$

Remark 1: We note from equation (4) that Z is affine in U . Hence, $M_T = Z^\top Z$ is quadratic in U , and maximizing (13) efficiently is challenging even with concavity assumptions on Φ .

In [16], the policy is chosen to optimize (13) with the A-optimal criterion. The inputs are restricted to periodic signals and are optimized in the frequency domain.

III. ONLINE GREEDY IDENTIFICATION

A simple and natural approach for system identification consists in updating the policy at each time step t in a greedy manner: the input u_t is chosen with energy γ^2 so as to maximize a one-step-ahead objective. Then, a new observation x_t is collected and the process repeats. In the formalism of Section I-D, this corresponds to the schedule $t_i = i$.

A. One-step-ahead objective

Following Section II-B, we adopt the optimal design functional $F = F_\Phi$. Upon choosing u_t , the policy π_t should maximize the design criterion Φ applied on the one-step-ahead, u_t -dependent information matrix, the past trajectory $x_{0:t}$ being fixed. The one-step-ahead information matrix is $M_t + \mathbb{E}_{\theta_s}[z_t z_t^\top]$, because the next u_t -dependent covariate is z_t . Therefore, one-step-ahead planning yields the following optimization problem:

$$\begin{aligned} \max_{u \in \mathbb{R}^m} \quad & \Phi \left(M_t + z(u)z(u)^\top \right) \\ \text{such that} \quad & \|u\|^2 \leq \gamma^2, \quad z(u) = \begin{pmatrix} x_t \\ u \end{pmatrix}. \end{aligned} \quad (14)$$

Remark 2: With this greedy policy, the energy constraint imposed for the choice of one input ensures that the global power constraint (2) is met.

The corresponding identification process is detailed in Algorithm 2. We will see in Section III-B that the D-optimality version of problem (14) can be solved accurately and at a low cost. Moreover, Algorithm 2 has the advantage of improving the knowledge of θ_* at each time step, so that the planning objective is constantly corrected. In this way, the bias introduced by the uncertainty about θ_* is minimized, whereas a large bias could impair the identification when planning over larger time sequences.

Algorithm 2 Greedy system identification

inputs initial guess θ_0 , noise variance σ^2 , power γ^2 , time horizon T , design criterion Φ
output final estimate θ_T
for $0 \leq t \leq T - 1$ **do**
 $u_t \in \operatorname{argmax}_{\|u\|^2 \leq \gamma^2} \Phi \left(\bar{M}_t + z(u)z(u)^\top \right)$
 play u_t , observe x_{t+1}
 $M_{t+1} = M_t + z_t z_t^\top$
 $\theta_{t+1} = M_{t+1}^{-1} (M_t \theta_t + z_t y_t)$
end for

B. Solving the one-step D-optimal design problem

We show that one-step-ahead D-optimal planning for online system identification is equivalent to a quadratic optimization problem that can be solved efficiently.

Proposition 4: For D-optimality, there exists a symmetric matrix $Q \in \mathbb{R}^{m \times m}$ and $b \in \mathbb{R}^m$ such that the problem (14) is equivalent to

$$\begin{aligned} \min_{u \in \mathbb{R}^d} \quad & u^\top Q u - 2b^\top u \\ \text{such that} \quad & \|u\|^2 \leq \gamma^2. \end{aligned} \quad (15)$$

Proof: By the matrix determinant lemma, we find that

$$\begin{aligned} \log \det (M_t + z(u)z(u)^\top) &= \log \det M_t \\ &\quad + \log (1 + z(u)^\top M_t^{-1} z(u)). \end{aligned} \quad (16)$$

Maximizing this quantity with respect to u is equivalent to maximizing $z(u)^\top M_t^{-1} z(u)$. The matrix M_t^{-1} is symmetric because M_t is symmetric, and so are its diagonal submatrices. Given the affine dependence of z in u and the (possible) block structure of z and M_t , $z(u)^\top M_t^{-1} z(u)$ is of the form $u^\top Q u - 2b^\top u$, up to a constant. We provide an explicit formula for Q and b in the case where $\theta = A$ in Section III-C ■

We now characterize the minimizers of Problem (15). We focus on maximal energy greedy policies ($\|u_t\|_2^2 = \gamma^2$) and thus consider the equality constrained problem

$$\begin{aligned} \min_{u \in \mathbb{R}^d} \quad & u^\top Q u - 2b^\top u \\ \text{such that} \quad & \|u\|^2 = \gamma^2. \end{aligned} \quad (17)$$

Proposition 5: Let us denote u_* a minimizer of (17), $\{\alpha_i\}$ the eigenvalues of Q , and u_i and b_i the coordinates of u_* and b in a corresponding orthonormal basis. Then there exists a nonzero scalar μ such that

$$u_i = b_i / (\alpha_i + \mu) \quad \text{and} \quad \sum_i \frac{b_i^2}{(\alpha_i + \mu)^2} = \gamma^2. \quad (18)$$

Proof: By the Lagrange multiplier theorem, there exists a nonzero scalar μ such that $Q u_* - b = -\mu u_*$, where μ can be scaled such that $Q + \mu I_d$ is nonsingular. Inverting the optimal condition and expanding the equality constraint yields the two conditions. ■

It follows from Proposition 5 that our greedy planning problem (15) can be solved at the cost of a scalar root-finding search and an eigenvalue decomposition. In [18], bounds are provided so as to efficiently initialize the root-finding method.

C. Known B_*

In the case where $\theta = A$, the next u_t -dependent covariate upon choosing u_t is x_{t+1} so one-step-ahead planning takes the form

$$\begin{aligned} \max_{u \in \mathbb{R}^m} \quad & \Phi \left(M_{t+1} + \sigma^2 I_d + x(u)x(u)^\top \right) \\ \text{such that} \quad & \|u\|^2 \leq \gamma^2, \quad x(u) = A_t x_t + B_* u. \end{aligned} \quad (19)$$

For D-optimality, we obtain a quadratic problem of the same form as (15), with the following matrices:

$$Q = -B^\top M_t^{-1} B_*, \quad b = B^\top M_t^{-1} A_t x_t. \quad (20)$$

IV. GRADIENT-BASED PLANNING

In this section, we propose a gradient-based approach to planning.

A. Gradient-based optimal design

In the adaptive identification scheme of Algorithm 1, the cost functions (3) and (13) can be optimized by projected gradient descent. Gradients with respect to U can be derived analytically (see [3], section 6 for the derivation of an adjoint equation) or automatically in an automatic differentiation framework. We propose a gradient-based planning method that consists in performing gradient descent directly on U in functional (13). Note that we optimize the inputs directly in the time domain, whereas other approaches such as [16] perform optimization in the frequency domain by restricting the control to periodic inputs. We rescale U at each step to ensure that the power constraint (2) is met. The t_i are chosen arbitrarily. The computational complexity of the algorithm is linear in T : each gradient step backpropagates through the planning time interval.

B. Gradient through the MSE oracle

Given the true parameters $\theta_\star = (A_\star B_\star)$, the optimal control for the MSE minimizes the MSE cost (3), as explained in Example 3. However, the dependance between Z and W makes this functional complicated to evaluate and to minimize with respect to the inputs, even when the true parameters θ_\star are known. We propose a numerical method to minimize (3) using Monte-Carlo sampling. One can sample a batch of b noise matrices $W_1, \dots, W_b \sim \mathcal{N}(0, \sigma^2 I)$, then approximate the gradient of (3) by

$$\nabla \text{MSE}(U) \simeq \frac{1}{b} \sum_{i=1}^b \nabla_U \text{tr} \left[Z(Z^\top Z)^{-2} Z^\top W_i W_i^\top \right], \quad (21)$$

and perform projected gradient descent. Although we have no convergence guarantees due to the complicated structure of the objective function, gradient descent does converge in practice, to a control that outperforms the adaptive controls.

V. PERFORMANCE STUDY

We compare our greedy algorithm to the gradient algorithms of Section IV and to the TOPLE algorithm of [16].

A. Complexity analysis

Definition 6 (Performance): The performance of the policy π is measured by the average error over the experiments on the true system: $\varepsilon = \text{MSE}(\pi)$. We study the performance of our algorithms as a function of the number of observations T and the computational cost C . We also introduce the computational rate $c = C/T$.

Algorithm 2 and the gradient identification algorithm have linear time complexity. Hence, we define c_{greedy} and c_{gradient} for a given number of gradient iterations. In practice, we find that $c_{\text{greedy}} \ll c_{\text{gradient}}$, where c_{gradient} is the computational rate needed for the gradient descent to converge. Furthermore, we expect the squared error to scale approximately as $1/T$ with the number of observations [19] (see [16]

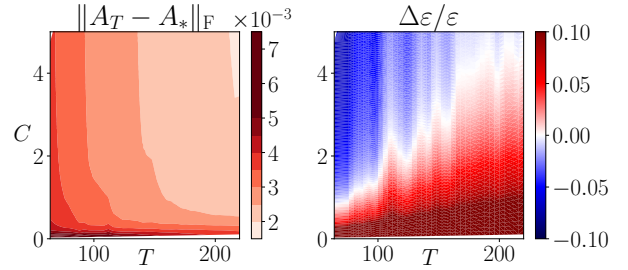


Fig. 1: Experimental (T, C) diagram. **Left** Performance of the gradient algorithm, with variable T and C (varying number of gradient steps). **Right** Relative performance of the gradient algorithm compared with the greedy algorithm: negative means that the gradient performs better.

Theorem 2.1). Given the above observations, we postulate that the performance of our algorithms takes the form

$$\varepsilon(C, T) = \varepsilon(c)/T, \quad (22)$$

which is experimentally verified. We build an experimental diagram where we plot the average estimation error for $\theta_\star = A_\star$ as a function of the two resources T and C for the gradient algorithm. Increasing C allows for more gradient steps. We run experiments with random matrices A_\star of size $d = 4$, with $B_\star = I_d$. We set $\gamma = 1$, $\sigma = 10^{-2}$, $T \in [60, 220]$. The gradient algorithm optimizes the D-optimality functional (13) with $\{t_i\} = \{0, 10, T/2, T\}$. The results obtained are to be compared with those of the greedy algorithm, which has a fixed, small computational rate c .

Our diagrams are plotted on Fig. 1. They show that the greedy algorithm is preferable in a phase of low computation rate: $C < c \times T$, as suggested by (22). The phase separation corresponds to a relatively high number of gradient steps. Indeed, the iso-performances along this line are almost vertical, meaning that the gradient descent has almost converged. Furthermore, the maximum performance gain of the gradient algorithm over the greedy algorithm is 10%.

B. Average estimation error

We now test the performances of our algorithms on random matrices, with the same settings as in the previous experiment. For each matrix A_\star , we also compute an oracle optimal control using the Monte-Carlo approach of Section IV-B with a batch size of $b = 100$, and run a random input baseline (see Example 1), as well as the TOPLE algorithm of [16].

The performances over time are shown in Figure 2. Both the gradient algorithm and the greedy algorithm closely approach the oracle. The former performs slightly better than the latter in average. However, the computational cost of the gradient algorithm is much larger, as shown in Table I. The values of c for the gradient-based algorithms are those found to achieve convergence in this setting (~ 100 for the gradient-based optimal design, ~ 1000 for TOPLE).

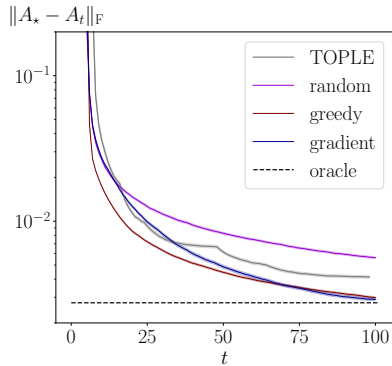


Fig. 2: Identification error for random A_* , averaged over 1000 samples.

TABLE I: Average computational rate for the different algorithms.

	Random	TOPLE [15]	MSE Gradient	Greedy
c	1	100	50	2.36

TABLE II: Frobenius error for A_* in the lateral system of the aircraft, $T = 150$. Our oracle algorithm reaches an error of 8.0×10^{-2} . The computational time is expressed in an arbitrary unit.

	Random	TOPLE [15]	MSE gradient	Greedy
Error	1.1×10^{-1}	8.6×10^{-2}	8.3×10^{-2}	8.2×10^{-2}
Time	1	55.7	25	1.13

C. Identification of an aircraft system

We now study a more realistic setting from aeronautics: we apply system identification to an aircraft system. We use the numerical values issued in a NASA report [4]. The lateral motion of a Lockheed Jet star is described by the slideslip and roll angles and the roll and yaw rates $(\beta, \phi, p, r)^\top := x$. The control variables are the aileron and rudder angles $(\delta_a, \delta_r)^\top := u$. The linear dynamics for an aircraft flying at 573.7 meters/sec at 6.096 meters are given by the following matrices obtained after discretization and normalization of the continuous-time system described in [4]:

$$A_* = \begin{pmatrix} .955 & -.0113 & 0 & -.0284 \\ 0 & 1 & .0568 & 0 \\ -.25 & 0 & -.963 & .00496 \\ .168 & 0 & -.00476 & -.993 \end{pmatrix}, \quad (23)$$

$$B_* = 0.1 \times \begin{pmatrix} 0 & 0.0116 \\ 0 & 0 \\ 1.62 & .789 \\ 0 & -.87 \end{pmatrix}, \quad (24)$$

with $\sigma = 1$ and $\gamma \simeq 4$ deg. We apply our algorithms on this LTI system. Our results are summarized in Table II.

The greedy algorithm estimates the system parameters as well as the gradient-based algorithms, but with substantially lighter computational cost. With small T , the estimate of A_* is too inaccurate for long-term planning to be effective.

It is more efficient to update the estimate and the policy frequently. We obtain similar results for the longitudinal system of a C-8 Buffalo aircraft [4].

VI. CONCLUSION

In this work, we explore a setting for linear system identification with strong constraints on the number of interactions with the real system and on the computing resources used for planning and estimation. We introduce an online greedy algorithm requiring minimal computing resources and show empirically that in a regime where the number of interactions with the system is small, it can actually outperform more sophisticated gradient-based methods. Extending this approach to optimal control for the LQR is an interesting direction for future research.

REFERENCES

- [1] Lennart Ljung. System identification. In *Signal analysis and prediction*, pages 163–173. Springer, 1998.
- [2] H.G. Natke. System identification: Torsten söderström and petre stoica. *Automatica*, 28(5):1069–1071, 1992.
- [3] G.C. Goodwin and R.L. Payne. *Dynamic System Identification: Experiment Design and Data Analysis*. Developmental Psychology Series. Academic Press, 1977.
- [4] NK Gupta, RK Mehra, and WE Hall Jr. Application of optimal input synthesis to aircraft parameter identification, 1976.
- [5] Thomas M. Moerland, Joost Broekens, and Catholijn M. Jonker. Model-based reinforcement learning: A survey, 2021.
- [6] Michel Gevers, Xavier Bombois, Roland Hildebrand, and Gabriel Solari. Optimal Experiment Design for Open and Closed-loop System Identification. *Communications in Information and Systems (CIS)*, 11(3):197–224, 2011. Special Issue Dedicated to Brian Anderson on the Occasion of His 70th Birthday: Part II.
- [7] V.V. Fedorov, V.V. Fedorov, W.J. Studden, E.M. Klimko, and Academic Press (London). *Theory of Optimal Experiments*. Cellular Neurobiology. Academic Press, 1972.
- [8] Friedrich Pukelsheim. *Optimal design of experiments*. SIAM, 2006.
- [9] L. Keviczky. "Design of Experiments" for the Identification of Linear Dynamic Systems. *Technometrics*, 17(3):303–308, 1975.
- [10] R. Mehra. Optimal inputs for linear system identification. *IEEE Transactions on Automatic Control*, 19(3):192–200, 1974.
- [11] Yassir Jedra and Alexandre Proutiere. Finite-time identification of stable linear systems optimality of the least-squares estimator. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 996–1001. IEEE, 2020.
- [12] Max Simchowitz, Horia Mania, Stephen Tu, Michael I Jordan, and Benjamin Recht. Learning without mixing: Towards a sharp analysis of linear system identification. In *Conference On Learning Theory*, pages 439–473. PMLR, 2018.
- [13] Tuhin Sarkar and Alexander Rakhlin. Near optimal finite time identification of arbitrary linear dynamical systems. In *International Conference on Machine Learning*, pages 5610–5618. PMLR, 2019.
- [14] Anastasios Tsiamis and George J Pappas. Finite sample analysis of stochastic system identification. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 3648–3654. IEEE, 2019.
- [15] Andrew Wagenmaker and Kevin Jamieson. Active learning for identification of linear dynamical systems. In *Conference on Learning Theory*, pages 3487–3582. PMLR, 2020.
- [16] Andrew J Wagenmaker, Max Simchowitz, and Kevin Jamieson. Task-optimal exploration in linear dynamical systems. In *International Conference on Machine Learning*, pages 10641–10652. PMLR, 2021.
- [17] David M Steinberg and William G Hunter. Experimental design: review and comment. *Technometrics*, 26(2):71–97, 1984.
- [18] William W Hager. Minimizing a quadratic over a sphere. *SIAM Journal on Optimization*, 12(1):188–208, 2001.
- [19] Stephen Boyd and Lieven Vandenberghe. *Introduction to applied linear algebra: vectors, matrices, and least squares*. Cambridge university press, 2018.